

May 31, 00 10:01	Main.m3	Page 1/1
5	<pre>MODULE Main ; IMPORT IO ; FROM Aufgabe35 IMPORT Random ; (* Bestimmung des Medians einer Zahlenfolge nach einer Divide-and-Conquer-Strategie. Idee: 1. Partitioniere das Feld genau wie bei QuickSort. 2. Je nachdem ob das Pivot-Element links oder rechts von der Mitte des Feldes zu liegen kommt, durchsuche rekursive entweder das linke *oder* das rechte Teilfeld. *) PROCEDURE Median(VAR a : ARRAY OF INTEGER) : INTEGER =</pre>	
10		
15	<pre> VAR m : INTEGER ; (* Position des Medians im sortierten Feld *) PROCEDURE Partition(l, r : INTEGER) : INTEGER = VAR v, t, i, j : INTEGER ; BEGIN i := l - 1 ; j := r ; v := a[r] ; (* waehle Pivot-Element *) REPEAT REPEAT INC(i) UNTIL a[i] >= v ; REPEAT DEC(j) UNTIL a[j] <= v ; t := a[i] ; a[i] := a[j] ; a[j] := t ; UNTIL j <= i ; (* Zeiger kreuzen *) a[j] := a[i] ; a[i] := a[r] ; a[r] := t ; RETURN i ; END Partition ; PROCEDURE Med(l, r : INTEGER) : INTEGER = VAR i : INTEGER ; BEGIN IF r > l THEN i := Partition(l, r) ; IF i = m THEN RETURN a[i] ; ELSIF i > m THEN RETURN Med(l, i-1) ; ELSE RETURN Med(i+1, r) ; END ; ELSE RETURN a[m] ; END ; END Med ; BEGIN a[0] := 0 ; (* Sentinel *) m := LAST(a) DIV 2 ; RETURN Med(l, LAST(a)) ; END Median ;</pre>	
50		
55	<pre>CONST n = 50 ; VAR data : ARRAY [0..n] OF INTEGER ; BEGIN FOR i := 1 TO n DO data[i] := Random(1000) ; IO.PutInt(data[i] ; IO.Put("\n") ; END ; IO.Put("Median: ") ; IO.PutInt(Median(data)) ; IO.Put("\n") ; END Main .</pre>	
65		

May 30, 00 10:28	output.txt	Page
963		
526		
992		
774		
5		
939		
53		
589		
338		
71		
10		
150		
270		
402		
263		
665		
15		
452		
881		
558		
695		
779		
20		
771		
581		
951		
132		
866		
25		
807		
821		
355		
180		
263		
30		
567		
142		
785		
153		
883		
35		
797		
704		
345		
6		
270		
40		
166		
493		
972		
366		
750		
92		
45		
775		
508		
704		
364		
50		
883		
Median : 558		